



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **analog signal annotation label**Found **22,483** of **151,219**

Sort results by

Display results


[Save results to a Binder](#)

[Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)
Results 141 - 160 of 200 Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) **8** [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**141** [Compiler transformations for high-performance computing](#)

David F. Bacon, Susan L. Graham, Oliver J. Sharp

December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4Full text available: [pdf\(6.32 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

142 [Integrating communication, cooperation, and awareness: the DIVA virtual office environment](#)

Markus Sohlenkamp, Greg Chwelos

October 1994 **Proceedings of the 1994 ACM conference on Computer supported cooperative work**Full text available: [pdf\(1.60 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

DIVA, a novel environment for group work, is presented. This prototype virtual office environment provides support for communication, cooperation, and awareness in both the synchronous and asynchronous modes, smoothly integrated into a simple and intuitive interface which may be viewed as a replacement for the standard graphical user interface desktop. In order to utilize the skills that people have acquired through years of shared work in real offices, DIVA is modeled after the standard of ...

Keywords: CSCW, awareness, groupware, integration, synchronous/asynchronous, virtual office

143 [Information extraction as a basis for high-precision text classification](#)

Ellen Riloff, Wendy Lehnert

July 1994 **ACM Transactions on Information Systems (TOIS)**, Volume 12 Issue 3Full text available: [pdf\(2.79 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


We describe an approach to text classification that represents a compromise between traditional word-based techniques and in-depth natural language processing. Our approach uses a natural language processing task called "information extraction" as a basis for high-precision text classification. We present three algorithms that use varying amounts of extracted information to classify texts. The relevancy signatures algorithm uses linguistic phrases; the a ...


Keywords: information extraction, text classification

144 Hidden understanding models of natural language

Scott Miller, Robert Bobrow, Robert Ingria, Richard Schwartz

June 1994 **Proceedings of the 32nd conference on Association for Computational Linguistics**

Full text available:  [pdf\(678.32 KB\)](#)

 [Publisher Site](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We describe and evaluate hidden understanding models, a statistical learning approach to natural language understanding. Given a string of words, hidden understanding models determine the most likely meaning for the string. We discuss 1) the problem of representing meaning in this framework, 2) the structure of the statistical model, 3) the process of training the model, and 4) the process of understanding using the model. Finally, we give experimental results, including results on an ARPA evalu ...

145 Tools and transformations—rigorous and otherwise—for practical database design

Arnon Rosenthal, David Reiner

June 1994 **ACM Transactions on Database Systems (TODS)**, Volume 19 Issue 2

Full text available:  [pdf\(3.19 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We describe the tools and theory of a comprehensive system for database design, and show how they work together to support multiple conceptual and logical design processes. The Database Design and Evaluation Workbench (DDEW) system uses a rigorous, information-content-preserving approach to schema transformation, but combines it with heuristics, guess work, and user interactions. The main contribution lies in illustrating how theory was adapted to a practical system, and how the consistency ...

Keywords: applications of database theory, computer-aided software engineering, data model translation, database design, database equivalence, design heuristics, entity-relationship model, heuristics, normalization, view integration

146 Parallel programming with control abstraction

Lawrence A. Crowl, Thomas J. LeBlanc

May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 3

Full text available:  [pdf\(3.68 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Parallel programming involves finding the potential parallelism in an application and mapping it to the architecture at hand. Since a typical application has more potential parallelism than any single architecture can exploit effectively, programmers usually limit their focus to the parallelism that the available control constructs express easily and that the given architecture exploits efficiently. This approach produces programs that exhibit much less parallelism that exists in the applic ...

Keywords: architectural adaptability, closures, control abstraction, data abstraction, early reply, multiprocessors, parallel programming languages, performance tuning

A general, fine-grained, machine independent, object-oriented language


Birger Andersen

May 1994 **ACM SIGPLAN Notices**, Volume 29 Issue 5Full text available:  [pdf\(653.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This paper introduces the general-purpose object-oriented programming language Ellie which supports machine independent fine-grained objects and parallelism. As something particular, classes, types, blocks, and methods are abstracted by first class objects/citizens called Ellie objects. Ellie demonstrates new approaches for abstraction and code reuse in parallel programming. The goals of Ellie have been to obtain an extremely flexible, machine independent, parallel language. Ellie tries to meet t ...

148 Evaluating hypermedia and learning: methods and results from the Perseus Project

Gary Marchionini, Gregory Crane

January 1994 **ACM Transactions on Information Systems (TOIS)**, Volume 12 Issue 1Full text available:  [pdf\(2.57 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Perseus Project has developed a hypermedia corpus of materials related to the ancient Greek world. The materials include a variety of texts and images, and tools for using these materials and navigating the system. Results from a three-year evaluation of Perseus use in a variety of college settings are described. The evaluation assessed both this particular system and the application of the technological genre to information management and to learning. The evaluation used a variety of methods ...

Keywords: human-computer interaction, hypermedia, learning, teaching

149 Cliché-based program editors


Richard C. Waters

January 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 1Full text available:  [pdf\(3.22 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

Keywords: abstract syntax tree schemas, computer-aided software engineering (CASE), plan diagrams, reuse

150 Analogical reasoning for knowledge discovery in a molecular biology database

Juergen Haas, Jeffrey S. Aaronson, G. Christian Overton

December 1993 **Proceedings of the second international conference on Information and knowledge management**Full text available:  [pdf\(1.28 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)151 A unification-based parser for relational grammar

David E. Johnson, Adam Meyers, Lawrence S. Moss

June 1993 **Proceedings of the 31st conference on Association for Computational Linguistics**Full text available:  [pdf\(535.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)
 [Publisher Site](#)

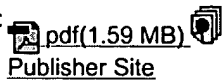
We present an implemented unification-based parser for relational grammars developed within the **stratified feature grammar (SFG)** framework, which generalizes Kasper-Rounds logic to handle relational grammar analyses. We first introduce the key aspects of SFG and a lexicalized, graph-based variant of the framework suitable for implementing relational grammars. We then describe a head-driven chart parser for lexicalized SFG. The

basic parsing operation is essentially ordinary feature-struct ...

152 Special issue on using large corpora: II: Coping with ambiguity and unknown words through probabilistic models

Ralph Weischedel, Richard Schwartz, Jeff Palmucci, Marie Meteer, Lance Ramshaw
June 1993 **Computational Linguistics**, Volume 19 Issue 2

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

[Publisher Site](#)

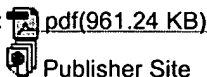
From spring 1990 through fall 1991, we performed a battery of small experiments to test the effectiveness of supplementing knowledge-based techniques with probabilistic models. This paper reports our experiments in predicting parts of speech of highly ambiguous words, predicting the intended interpretation of an utterance when more than one interpretation satisfies all known syntactic and semantic constraints, and learning caseframe information for verbs from example uses. From these experiments, w ...

153 Contributed papers: Tuples, discontinuity, and gapping in categorial grammar

Glyn Morrill, Teresa Solias

April 1993 **Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics**

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

[Publisher Site](#)

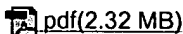
This paper solves some puzzles in the formalisation of logic for discontinuity in categorial grammar. A 'tuple' operation introduced in [Solias, 1992] is defined as a mode of prosodic combination which has associated projection functions, and consequently can support a property of unique prosodic decomposability. Discontinuity operators are defined model-theoretically by a residuation scheme which is particularly amenable proof-theoretically. This enables a formulation which both improves on th ...

154 The concurrency workbench: a semantics-based tool for the verification of concurrent systems

Rance Cleaveland, Joachim Parrow, Bernhard Steffen

January 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 1

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Concurrency Workbench is an automated tool for analyzing networks of finite-state processes expressed in Milner's Calculus of Communicating Systems. Its key feature is its breadth: a variety of different verification methods, including equivalence checking, preorder checking, and model checking, are supported for several different process semantics. One experience from our work is that a large number of interesting verification methods can be formulated as combinations of a small number ...

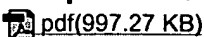
Keywords: automatic verification, concurrency, finite-state systems, process algebra

155 Verification of asynchronous interface circuits with bounded wire delays

Srinivas Devadas, Kurt Keutzer, Sharad Malik, Albert Wang

November 1992 **Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design**

Full text available:



Additional Information: [full citation](#), [references](#), [index terms](#)

156 VHDL 1076–1992 languages changes

Andrew Guyler

November 1992 **Proceedings of the conference on European design automation**

Full text available:  [pdf\(590.69 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

157 [Natural language processing \(NLP\) & hypermedia: Event relations at the phonetics/phonology interface](#)

Julie Carson-Berndsen, Dafydd Gibbon

August 1992 **Proceedings of the 14th conference on Computational linguistics - Volume 4**


Full text available:  [pdf\(338.89 KB\)](#) Additional Information: [full citation](#), [references](#)



158 [A program integration algorithm that accommodates semantics-preserving transformations](#)

Wuu Yang, Susan Horwitz, Thomas Reps

July 1992 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 1 Issue 3

Full text available:  [pdf\(3.07 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Given a program Base and two variants, A and B, each created by modifying separate copies of Base, the goal of program integration is to determine whether the modifications interfere, and if they do not, to create an integrated program that includes both sets of changes as well as the portions of Base preserved in both variants. Text-based integration techniques, such as the one used by the Unix dif ...

Keywords: coarsest partition, control dependence, data dependence, data-flow analysis, flow dependence, program dependence graph, program integration, program representation graph, static-single-assignment form



159 [PCCTS reference manual: version 1.00](#)

T. J. Parr, H. G. Dietz, W. E. Cohen

February 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 2


Full text available:  [pdf\(3.77 MB\)](#) Additional Information: [full citation](#), [citations](#), [index terms](#)



160 [Computer-assisted microanalysis of parallel programs](#)

Timothy J. Hickey, Jacques Cohen, Hirofumi Hotta, Thierry PetitJean

January 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 1

Full text available:  [pdf\(3.02 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

This paper consists of two parts: the first provides the theoretical foundations for analyzing parallel programs and illustrates how the theory can be applied to estimate the execution time of a class of parallel programs being executed on a MIMD computer. The second part describes a program analysis system, based on the theoretical model, which allows a user to interactively analyze the results of executing (or simulating the execution) of such parallel programs. Several examples illustrate ...

Keywords: event graph, execution graph, execution trace, microanalysis, speed up



Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)